Many Roads to Synchrony: Natural Time Scales and Their Algorithms

Ryan G. James,[1,*] John R. Mahoney,[1,†] Christopher J. Ellison,[1,‡] and James P. Crutchfield[1,2,§]

[1] *Complexity Sciences Center and Physics Department, University of California at Davis, One Shields Avenue, Davis, CA 95616*
[2] *Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501*

(Dated: October 28, 2010)

We survey the variety of ways in which one synchronizes to a stochastic process. We define associated length scales, providing characterization theorems and efficient algorithms for their calculation. We demonstrate that many of the length scales are minimized by using the ϵ-machine, compared to all of a process's alternative models. We also show that the concept of Markov order, familiar in stochastic process theory, is a topological property of the ϵ-machine presentation. Moreover, we find that it can only be computed when using the ϵ-machine, not any alternative. We illustrate the results by presenting evidence that infinite Markov order and infinite crypticity are dominant properties in the space of finite-memory processes.

Keywords: Markov order, cryptic order, synchronization, ϵ-machine

PACS numbers: 02.50.-r 89.70.+c 05.45.Tp 02.50.Ey 02.50.Ga

I. INTRODUCTION

Stochastic processes are often described by the spatial and temporal length scales over which correlations exist. In physics, the range of correlations is a structural property, giving the range over which significant energetic coupling exists between the system's degrees of freedom. In time series analysis, knowing the length scale of correlations is key to successful forecasting. In biosequence analysis, the decay of correlations along DNA base pairs determines in some measure the difficulty faced by a replicating enzyme as it "decides" to begin copying a gene. The common element in these is that the correlation scale determines how quickly the observer (analyzer, forecaster, or enzyme) comes to *synchronize* to the process—that is, comes to know a relevant structure of the stochastic process.

We recently showed that there are a number of distinct, though related, length scales associated with synchronizing to presentations of stationary stochastic processes [1]. Here, we show that these length scales come in two flavors: *topological* and *statistical*. The topological length scales depend only on the underlying graph topology of the effective states and their transitions, while the statistical length scales vary with the underlying probabilistic structure.

Specifically, we investigate measures of synchronization and their associated lengths scales for hidden Markov models (HMMs)—a particular class of processes that have an internal (hidden) Markovian dynamic that produces an observed data sequence. We compare and contrast various notions of being synchronized in a setting where one is given a model and a sequence of observations. Several of the length scales described (e.g., Markov order and synchronization time) are familiar in fields such as physics, stochastic processes, information theory, and machine learning, while others (e.g., cryptic order) are relatively new and their applicability has yet to be fully appreciated. In this work, we provide efficient algorithms for their calculation.

The development proceeds as follows. Section II defines the length scales and related properties of interest for HMMs. Section III then introduces and proves several results that serve to relate them, give insight into their meaning, and make their computation tractable. Next, Sec. IV provides efficient methods for computing these length scales. By way of illustration, Sec. V presents a survey of Markov and cryptic orders among finite-state ϵ-machines—a unique and privileged representation of a process—which shows that infinite correlation is a dominant property in the space of processes and that it is generically hard to exactly synchronize to stationary processes. Finally, Sec. VI concludes by suggesting applications for which these algorithms will prove useful.

*Electronic address: rgjames@ucdavis.edu
†Electronic address: jrmahoney@ucdavis.edu
‡Electronic address: cellison@cse.ucdavis.edu
§Electronic address: chaos@cse.ucdavis.edu

II. BACKGROUND

We assume the reader has an introductory knowledge of information theory and finite-state machines, such as that found in the first few chapters of Ref. [2] and Ref. [3], respectively. Furthermore, we make use of ϵ-machines, a particular representation of a process that makes many properties directly and easily computable; see Ref. [4]. A cursory understanding of symbolic dynamics, such as that found in the first few chapters of Ref. [5], is useful for several of the results.

A. Processes

We denote subsequences in a time series as $X_{a:b}$, where $a < b$, to refer to the random variable sequence $X_a X_{a+1} X_{a+2} \cdots X_{b-1}$, which has length $b - a$. The *past* $\overleftarrow{X}$ is then $X_{-\infty:0}$ and the *future* $\overrightarrow{X}$ is $X_{0:\infty}$. We generally use w to refer to a *word*—a sequence of symbols drawn from an alphabet $\mathcal{A}$. We place two words, u and v, adjacent to each other to mean concatenation: $w = uv$. We define a *process* to be a joint probability distribution $\Pr(\overleftrightarrow{X})$ over $\overleftrightarrow{X} = \overleftarrow{X}\overrightarrow{X}$.

A *presentation* of a given process is any state-based representation that generates the process. A process's *ϵ-machine* is its unique, minimal unifilar presentation [15]. The recurrent states of a process's ϵ-machine are denoted $\mathcal{S}$ and the recurrent states of any other presentation are denoted $\mathcal{R}$ [16]. Distributions over states which occur infinitely often are denoted $\{\mathcal{R}\}$; these are the recurrent state distributions.

Following Ref. [1], we seek to give our definitions and algorithms flexibility by allowing generalizations of the following definitions to all of a process's non-ϵ-machine presentations. Fortunately, our definitions change only slightly. All asymptotically synchronizing presentations [6] allow essentially the same definitions, with causal states replaced by the presentation's states. Unifilar presentations with positive gauge information [1], however, require that the definitions replace states with asymptotically recurrent state-distributions. (The latter are the recurrent states of the mixed-state presentation; see Ref. [7].) For example, the synchronization order can never be finite in a presentation with positive gauge information, but if we consider the state distributions that are visited infinitely often as the "state" to which we synchronize, the definition retains its spirit. Nonunifilar presentations have no such simple generalization. We summarize the state of affairs in Table I.

For the remainder of this paper we use causal states $\mathcal{S}$ in all definitions, propositions, theorems, proofs, and algorithms for convenience. However, except where explicitly stated as applying only to ϵ-machines, these can be exchanged for the states of an alternative model according to Table I.

Presentation Class	Generalization Rule
Asymptotically Synchronizing	$\mathcal{S} \to \mathcal{R}$
Nonsynchronizing Unifilar	$\mathcal{S} \to \{\mathcal{R}\}$
Nonunifilar	Unknown or N/A

TABLE I: Brief guide to generalizing the synchronization definitions to wider classes of presentation.

B. Minimal Synchronizing Words

For the synchronization problem, we consider an observer who begins with a correct model (a presentation) of a process. The observer, however, has no knowledge of the process's internal state. The challenge is to analyze how an observer's knowledge of the internal state changes as more and more measurements are observed.

At first glance, one might say that the observer's knowledge should never decrease with additional measurements, corresponding to a never-increasing state uncertainty, but this is generically not true. In fact, it is possible for the observer's knowledge (measured in bits) to oscillate with each new measurement. The crux of the issue is that additional measurements are being used to inquire about the *current* state rather than the state at some fixed moment in time.

It is helpful to identify the set of words that take the observer from the condition of total ignorance to exactly knowing a process's state. First, we introduce what we mean by synchronization in terms of lack of state uncertainty. Second, we define the set of minimal synchronizing words.

Definition 1. *A word w of length L is* synchronizing *if the Shannon entropy over the internal state, conditioned on w, is zero:*

$$Sync(w) \quad \Leftrightarrow \quad H[\mathcal{S}_\ell | X_{0:\ell} = w] = 0 \ , \tag{1}$$

where $Sync(w)$ is Boolean function.

Definition 2. *A presentation's set of* minimal synchronizing words *is the set of synchronizing words that have no synchronizing prefix:*

$$\mathcal{L}_{\mathrm{sync}} \equiv \{w \mid Sync(w) \text{ and}, \\ \neg Sync(u) \text{ for all } u : w = uv\} \ . \tag{2}$$

Remark. $\mathcal{L}_{\mathrm{sync}}$ *is a prefix-free, regular language. If*

each word is associated with its probability of being observed, we obtain a prefix-free code encoding each path to synchrony—a word in $\mathcal{L}_{\text{sync}}$—with the associated probability of synchronizing via that path. These codes are generally nonoptimal in the familiar information-theoretic sense.

C. Markov Order

Of the length scales to be surveyed, a process's Markov order is probably the most widely studied. It measures the memory in a time series and roughly corresponds to the length of the longest correlations that exist.

Definition 3. *The* Markov order R *is the number of observations required to predict the future as accurately as if using the infinite past.*

$$R \equiv \min\{\ell \mid \Pr(\overrightarrow{X}|\overleftarrow{X}) = \Pr(\overrightarrow{X}|X_{-\ell:0})\} \ . \tag{3}$$

If this condition does not hold for any finite ℓ, the Markov order is ∞.

A further interpretation of the Markov order is as the length at which a process's *block-entropy curve* $H[X_{0:\ell}]$ reaches its asymptote [1]:

$$R = \min\{\ell \mid H[X_{0:\ell}] = \mathbf{E} + \ell h_\mu\} \ . \tag{4}$$

In short, R is the length of data one must see before being able to predict upcoming symbols with the minimum error rate.

D. Cryptic Order

The cryptic order [7, 8] has several differing interpretations, but is best thought of as the number of observations required to account for the asymptotic amount of state information not transmitted to the future.

Definition 4. *The* cryptic order k_χ *[9] is the number of presentation states that cannot be retrodicted given the infinite future:*

$$k_\chi \equiv \min\{\ell \mid H[\mathcal{S}_\ell|\overrightarrow{X}] = 0\} \ . \tag{5}$$

Once again, if this condition does not hold for any finite ℓ, then $k_\chi = \infty$.

This can also be seen as the length at which the *block-state entropy* $H[X_{0:\ell}, \mathcal{S}_\ell]$ reaches its asymptotic behavior [1]. We can therefore understand this as a model-centric parallel of the Markov order statement in Eq. (4):

$$k_\chi = \min\{\ell \mid H[X_{0:\ell}, \mathcal{S}_\ell] = \mathbf{E} + \ell h_\mu\} \ . \tag{6}$$

E. Synchronization Order

According to Sec. II B, one is synchronized to a process's presentation after seeing word w if there is complete certainty in the state. We now expand this view slightly to ask about synchronization over all words of a particular length. Equivalently, we examine synchronization to an ensemble of process realizations.

Definition 5. *The* synchronization order $k_{\mathbf{S}}$ *[1] is the minimum length for which every word is a synchronizing word:*

$$k_{\mathbf{S}} \equiv \min\{\ell \mid H[\mathcal{S}_\ell|X_{0:\ell}] = 0\} \ . \tag{7}$$

As for the Markov and cryptic orders, $k_{\mathbf{S}}$ is considered ∞ when the condition does not hold for any finite ℓ.

F. The Synchronization Distribution

Taking a slightly more general view than the synchronization order, we consider statistical properties of synchronization, rather than just the absolute length at which an ensemble will all be synchronized. In this vein, we define a distribution that gives the probability for a word to first synchronize at length ℓ.

Definition 6. *The* synchronization distribution S *gives the probability of synchronizing to a presentation at length ℓ:*

$$S(\ell) \equiv \sum_{w \in \mathcal{L}_{\text{sync}}} \Pr(w) \delta(|w| - \ell) \ . \tag{8}$$

where δ is the Kronecker delta function.

Remark. *S is normalized: $\sum_{\ell=0}^{\infty} S(\ell) = 1$.*

We now draw out two particular quantities from this distribution—quantities that have observable meaning for a presentation.

Definition 7. *The* synchronization time τ *[1, 10] is the average number of observations needed to synchronize to a presentation:*

$$\tau \equiv \sum_{w \in \mathcal{L}_{\text{sync}}} |w| \Pr(w) \tag{9}$$

$$= \mathbb{E}_\ell[S(\ell)] \ , \tag{10}$$

where the second equality shows that τ is also equal to the expectation value of the synchronization distribution.

The synchronization time is useful for understanding how long it takes *on average* to synchronize to a model.

This is as opposed to the Markov order which is the minimal longest-synchronization-time across all presentations of a process.

Definition 8. *The* synchronization entropy H_{sync} *is the uncertainty in the synchronization distribution:*

$$H_{\text{sync}} \equiv H[S(\ell)] \ . \tag{11}$$

Remark. *Note that this is quite distinct from the* synchronization information $\mathbf{S}$ *of Ref. [11]:*

$$\mathbf{S} = \sum_{\ell=1}^{\infty} H[\mathcal{S}_\ell | X_{0:\ell}] \ .$$

The synchronization entropy, in contrast, measures the flatness of the synchronization distribution. And, since the synchronization distribution decays exponentially with length, the fatter the tail, the higher the uncertainty in synchronization.

III. RESULTS

We now provide several results related to these length scales that shed light on their nature introducing connections and simplifications that make their computation tractable.

Proposition 1. *The synchronization order is:*

$$k_{\mathbf{S}} = \ max\{R, k_\chi\} \ . \tag{12}$$

Proof. *First, note that:*

$$H[\mathcal{S}_\ell | X_{0:\ell}] = H[X_{0:\ell}, \mathcal{S}_\ell] - H[X_{0:\ell}] \ . \tag{13}$$

Since the block-state entropy upper bounds the block entropy, the conditional entropy above can only reach its asymptotic value once both terms have individually reached their asymptotic behavior. The latter are controlled by k_χ and R, respectively. □

This result reduces the apparent diversity of length scales, eventually allowing one to calculate the Markov order via the synchronization order, which itself is directly computable.

Proposition 2. *For ϵ-machines:*

$$R = k_{\mathbf{S}} \ . \tag{14}$$

Proof. *Applying the causal equivalence relation $\sim_\epsilon$ to Def. 3 we find:*

$$\Pr(\overrightarrow{X} | \overleftarrow{X}) = \Pr(\overrightarrow{X} | X_{-\ell:0}) \implies \overleftarrow{X} \sim_\epsilon X_{-R:0} \ . \tag{15}$$

This further implies that the causal states $\boldsymbol{\mathcal{S}}$ are completely determined by $X_{-R:0}$:

$$H[\mathcal{S}_0 | X_{-R:0}] = 0 \ . \tag{16}$$

This statement is equivalent to the Markov criterion. □

Remark. *This provides an alternate proof that the cryptic order k_χ is bounded above by the Markov order R in an ϵ-machine via a simple shift in indices:*

$$H[\mathcal{S}_0 | X_{-R:0}] = 0 \tag{17}$$
$$\implies H[\mathcal{S}_R | X_{0:R}] = 0 \tag{18}$$
$$\implies H[\mathcal{S}_R | \overrightarrow{X}] = 0 \ . \tag{19}$$

This proposition gives indirect access to the Markov order via a particular presentation—the ϵ-machine. Since the Markov order is not defined as a property of a presentation it would generally be unobtainable, but due to unique properties of the ϵ-machine, it can be accessed through the synchronization order.

There is a subclass of ϵ-machines to which one synchronizes in finite time; these are the *exact* ϵ-machines of Ref. [12].

Proposition 3. *Given an exact ϵ-machine with finite Markov order R, the subshift of finite type that underlies it has a "step" [5] equal to R.*

Corollary 1. *Given an exactly synchronizing ϵ-machine, the underlying sofic system is a subshift of finite type if and only if R is finite.*

Remark. *A process with infinite Markov order can have a presentation whose underlying sofic system is a subshift of finite type.*

These results draw out a connection with length scales of sofic systems from symbolic dynamics. Subshifts of finite type have a probability-agnostic length scale analog of the Markov order known as the 'step" [5]. In the case of ϵ-machine presentations, they are in fact equal.

We will now show that two of the lengths defined—the cryptic and synchronization orders—are topological. That is, they are properties of the presentation's graph topology and are independent of transition probabilities so long as changes to the probabilities do not remove transitions and do not cause states to merge. Additionally, due to Prop. 2, the Markov order can be thought of as topological. All three are topological since they depend only on the length at which a conditional entropy vanishes, not on how it vanishes.

Theorem 1. *Synchronization order $k_{\mathbf{S}}$ is a topological property of a presentation.*

Proof. *Beginning from Def. 5, there is length $\ell = k_{\mathbf{S}}$ at which:*

$$H[\mathcal{S}_\ell | X_{0:\ell}] = \sum_{w \in \mathcal{A}^\ell} \Pr(w) H[\mathcal{S}_\ell | X_{0:\ell} = w] = 0 \ .$$

Thus, $H[\mathcal{S}_\ell | X_{0:\ell} = w] = 0$ for all $w \in \mathcal{A}^\ell$, the set of length-ℓ words with positive probability. Since every word of length ℓ is synchronizing, ℓ is certainly greater than the synchronization order. As synchronizing words are synchronizing regardless of their probability of occurring, the synchronization order $k_{\mathbf{S}}$ is topological. □

Corollary 2. *Markov order R is a topological property of an ϵ-machine.*

Proof. *Since $k_{\mathbf{S}}$ is a topological property by Thm. 1 and since an ϵ-machine's $R = k_{\mathbf{S}}$ by Prop. 2, the Markov order is topological.* □

Theorem 2. *Cryptic order k_χ is a topological property of a presentation.*

Proof. *Beginning from Def. 4, there is a length $\ell = k_\chi$ at which:*

$$\begin{aligned}
0 &= H[\mathcal{S}_\ell | \overrightarrow{X}] \\
&\overset{(1)}{=} \sum_{\overrightarrow{x} \in \mathcal{A}^\infty} \Pr(\overrightarrow{x}) H[\mathcal{S}_\ell | \overrightarrow{X} = \overrightarrow{x}] \\
&\overset{(2)}{=} \sum_{w \in \mathcal{L}_{\text{sync}}} \Pr(w, \sigma_w) H[\mathcal{S}_\ell | X_{0:|w|} = w, \mathcal{S}_{|w|} = \sigma_w] \ .
\end{aligned}$$

Here, step (1) simply expands the conditional entropy. Step (2) is true provided that the sum is over minimal synchronizing words and σ_w is the state to which one synchronizes via w. This final sum is zero only if the sum vanishes term-by-term. Thus, given a word that synchronizes and the state synchronized to, each term provides a cryptic-order candidate*—the number of states that could not be retrodicted from that state and word. Finally, the longest such cryptic order candidate is the cryptic order for the presentation.* □

Restated, the cryptic order k_χ is topological as it depends only on the minimal synchronizing words, which are topological by definition.

IV. ALGORITHMS

We are now ready to turn to the primary task—computing the various synchronization length scales given a presentation. While all of these algorithms have compute times that are exponential in the number of machine states, we find them to be very efficient in practice. This is particularly the case compared to naive algorithms to compute these properties. For example, computing synchronization, Markov, or cryptic orders by testing successively longer blocks of symbols is exponential in the length of the longest block tested. Worse, in the case of non-Markovian/∞-cryptic processes the naive algorithm will not halt. In addition, the naive implementation of Thm. 2 given in the proof to compute the cryptic order has a compute time of $O(2^{2^N})$, whereas the one presented below is a simple exponential of N.

Unsurprisingly, given the results provided in Sec. III, we begin with the minimal synchronizing words as they are the underpinnings of the synchronization and cryptic orders. The algorithms make use of standard procedures. Most textbooks on algorithms provide the necessary background; see, for example, Ref. [13].

A. Minimal Synchronizing Words

We construct a deterministic finite automaton (DFA) that recognizes $\mathcal{L}_{\text{sync}}$ of a given presentation $\mathcal{M} = (Q, E)$, where Q are the states and E are the edges. This is done as follows:

Algorithm 1.

1. *Begin with the recurrent presentation $\mathcal{M}$.*
2. *Construct $\mathcal{M}$'s power automaton $2^{\mathcal{M}}$, producing a DFA $\mathcal{T} = 2^{\mathcal{M}}$.*
3. *Set the node in $\mathcal{T}$ that corresponds to all $\mathcal{M}$'s states as $\mathcal{T}$'s start state.*
4. *Remove all edges between singleton states of $\mathcal{T}$. (These are the edges from $\mathcal{M}$.)*
5. *Set all singleton states of $\mathcal{T}$ as accepting states.*

Now, we enumerate $\mathcal{L}_{\text{sync}}$ via an ordered breadth-first traversal of $\mathcal{T}$, outputting each accepted word.

B. Synchronization Order

Thanks to Eq. (1) we see that $k_{\mathbf{S}}$ is the shortest length ℓ that encompasses all of $\mathcal{L}_{\text{sync}}$. This is, trivially, the longest word in $\mathcal{L}_{\text{sync}}$. With this, computing the synchronization order reduces to:

Algorithm 2.

1. *If $\mathcal{L}_{\text{sync}}$ is infinite, return ∞.*

2. *Enumerate each word in $\mathcal{L}_{\text{sync}}$ and return the length of the longest word.*

The test in the first step can be done simply by running a loop-detection algorithm on DFA $\mathcal{T}$. If there is a loop, then $\mathcal{L}_{\text{sync}}$ is infinite.

C. Markov Order

Due to Thm. 2, a process's Markov order can be computed by finding the synchronization order of the process's ϵ-machine. If one does not have the ϵ-machine for a process, but rather some other unifilar presentation, it is still possible in some cases to obtain the Markov order through the synchronization order. That is, the algorithms for $k_{\mathbf{S}}$ and k_χ provide probes into the presentation's length scales. It can be the case that R is accessible to those probes, if $k_\chi < k_{\mathbf{S}}$, but it is only guaranteed to be accessible in the case of ϵ-machines. Note, there exist techniques for constructing the ϵ-machine from any presentation [7].

D. Cryptic Order

The cryptic-order candidates of Thm. 2 need not be computed individually; they can be computed in parallel by modifying the topological transient structure provided by the power automaton. One then retrodicts on the entire graph structure. In the following algorithm $\mathcal{T}$ refers to the power automata of the machine $\mathcal{M}$. The states p, q, and r of $\mathcal{T}$ are elements of the power set of the states of $\mathcal{M}$. By the *predecessors* of a state q along edge $p \xrightarrow{x} q$ we refer to the set $p' = \{m | (m \xrightarrow{x} n) \in \mathcal{M}$ and $m \in p$ and $n \in q\}$. These are the states $m \in p$ which actually transition to a state $n \in q$ on symbol x. By *subset construction* below we refer to the standard NFA-to-DFA conversion algorithm [3].

Algorithm 3.

1. *Construct the power automaton $\mathcal{T} = 2^{\mathcal{M}}$.*
2. *Add each edge $p \xrightarrow{x} q$ in $\mathcal{T}$ to a queue.*
3. *For each edge $p \xrightarrow{x} q$ in the queue:*
   (a) *If edge has already been processed:*
      i. *Restart loop with next edge in the queue.*
   (b) *Find the predecessors p' of q along $p \xrightarrow{x} q$.*
   (c) *If $|p'| > 1$ and $p' \neq p$:*
      i. *Remove edge $p \xrightarrow{x} q$.*
      ii. *Perform the subset construction on p' (implicitly adds the edge $p' \xrightarrow{x} q$).*
      iii. *For each $r \xrightarrow{y} p$:*
         A. *Add edge $r \xrightarrow{y} p'$.*
      iv. *Mark $p \xrightarrow{x} q$ and $p' \xrightarrow{x} q$ as processed.*
   (d) *Add all edges created in steps 3.c.ii and 3.c.iii.A to the queue.*

The result is an automaton. The longest path in it from a transient state to a recurrent is the cryptic order. Skipping previously processed edges is important since for some topologies the algorithm can enter a cycle where it will remove and then later add the same edge, ad infinitum.

This algorithm for computing the cryptic order only holds for unifilar presentations.

E. The Synchronization Distribution

There are two methods to compute the synchronization distribution. The first requires an ϵ-machine with finite recurrent and transient components. The second, only a finite recurrent component. We present the former case first.

Algorithm 4.

1. *Perform an ordered breadth-first traversal of the finite ϵ-machine.*
2. *While traversing, keep track of the word induced by the path and the product of the probabilities along that path.*
3. *When a recurrent node is reached, stop that particular thread of the traversal.*
4. *Sum the probabilities of all words with the same length.*

This algorithm produces each minimal synchronizing word and its probability in lexicographic order. Then words of each length can be grouped and their probabilities summed to get the synchronization distribution.

The second algorithm is used when an ϵ-machine with finite transient structure is not available:

Algorithm 5.

1. *Produce all the minimal synchronizing words from the DFA given in Algorithm 1.*
2. *For each minimal synchronizing word, compute its probability using the recurrent ϵ-machine [7].*

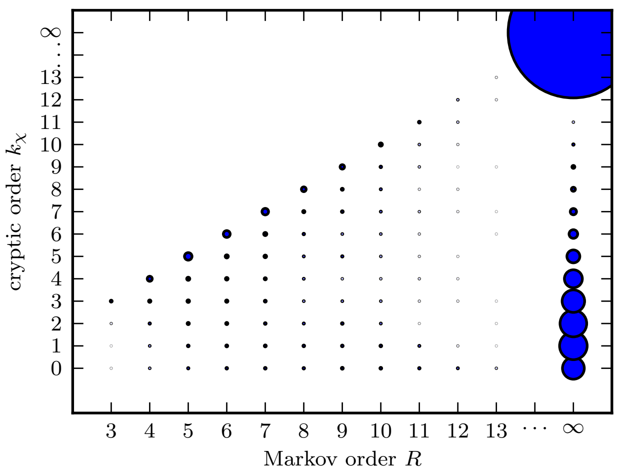

FIG. 1: The distribution of Markov order R and cryptic order k_χ for all $1,132,613$ six-state, binary-alphabet, exactly-synchronizing ϵ-machines. The marker size is proportional to the number of ϵ-machines within this class at the same (R, k_χ) value.

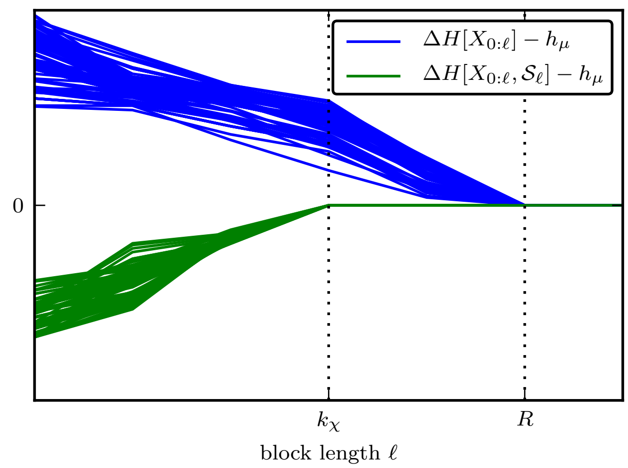

FIG. 2: Entropy convergence curves versus block length ℓ for a family of processes with the same topology, but different transition probabilities. The each linear asymptotic behavior subtracted out of each curve (see inset). The Markov and cryptic orders (the lengths at which the blue (dark) and green (lighter) lines, respectively, become flat) are independent of the selected probabilities.

3. *Sum the probabilities of all words with the same length.*

Once the distribution is computed using one of the above algorithms, it is trivial to compute the Shannon entropy and mean of the distribution to get the synchronization entropy and synchronization time, respectively.

V. GENERICITY OF SYNCHRONIZATION

We illustrate applying the above results and algorithms by empirically answering several simple questions. How typical are infinite Markov order and infinite cryptic order presentations in the space of finite-state processes?

Restricting ourselves to ϵ-machines, we enumerate all binary processes with a given number of states to which one can exactly synchronize [14]. Using these ϵ-machines one can compute, as we just showed, their Markov and cryptic orders. The result for all of the $1,132,613$ six-state ϵ-machines is shown in Fig. 1.

The number of ϵ-machines that share a (R, k_χ) pair is illustrated by the size of the circle at that (R, k_χ). It is easily seen that the vast majority of processes—in fact, 98%—are non-Markovian at this state-size (6). Furthermore, most (85% to be exact) of those non-Markovian processes are also ∞-cryptic. One concludes that, in the space of finite-state processes, infinite-range correlation and infinitely diffuse internal state information are the overwhelming rule. As a consequence, it is generically difficult to synchronize.

Also of interest are the "forbidden" (R, k_χ) pairs within the space of 6-state ϵ-machines. For example, ϵ-machines with $k_\chi = 4, 5, 8, 10, 11$ do not occur with $R = 13$. In addition, in the case of infinite Markov order, but finite cryptic order, the latter appears to be bounded above by $k_\chi = 11$ despite the fact that larger finite cryptic orders exist for finite Markov order processes.

Since Markov and cryptic orders are topological, they are independent of the probabilities assigned to edges in a presentation. This can be seen quite dramatically by assigning random probabilities to the edges in a presentation and plotting the resulting block and block-state entropies (actually, their scaled derivatives). The result is shown in Fig. 2. Note that, over a range of randomly sampled transition probabilities and therefore over a range of stochastic processes, convergence occurs at the exactly same lengths.

Finally, we survey the distribution of synchronization times τ and synchronization entropies H_{sync} for all $1,388$ four-state, binary-alphabet, exact ϵ-machines with uniform outgoing transition probabilities. See Fig. 3. It is interesting to note that there is structure in the distribution in the form of veils. However, the veils are not the entirety of the distribution, there are many machines that fall elsewhere.

VI. CONCLUSION

We demonstrated how to compute a number of length scales for hidden Markov models, most of which can be

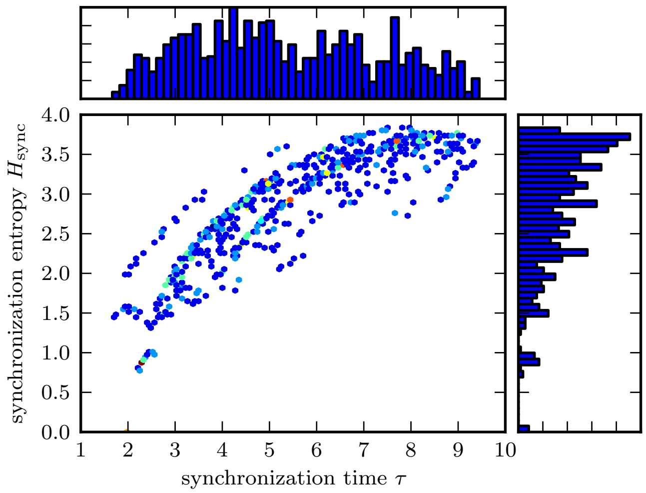

FIG. 3: Distribution of synchronization time τ and synchronization entropy H_{sync} for all $1,388$ four-state, binary-alphabet, exactly-synchronizing ϵ-machines with uniform outgoing transition probabilities. Individual histograms for each property are shown above and to the right.

motivated in terms of a process's, or its associated presentations', synchronization properties. Interestingly, we found that one of the most fundamental and important properties—the Markov order R—is computable only using the process's ϵ-machine. While other properties can be computed using other presentations, they are typically minimized when the presentation used is the ϵ-machine. In addition, the ϵ-machine provides an exact method for computing the Markov order whereas, in general, the Markov order can only be estimated. It is important to stress however that this is not a silver bullet. Given a time series, the ϵ-machine must still be estimated to compute the Markov order. Often, however, one can analytically calculate a process's ϵ-machine. In this case, the preceding gives exact results, as illustrated in our survey of Markov and cryptic orders for processes described by six-state ϵ-machines.

Acknowledgments

We thank Ben Johnson and Nick Travers for many invaluable discussions. The work was partially supported by the Defense Advanced Research Projects Agency (DARPA) Physical Intelligence project. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the DARPA or the Department of Defense.

[1] J. P. Crutchfield, C. J. Ellison, Ryan G. James, and J. R. Mahoney. Synchronization and control in intrinsic and designed computation: An information-theoretic analysis of competing models of stochastic computation. *CHAOS*, 20(3):037105, 2010.

[2] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, 1991.

[3] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, 1979.

[4] C. R. Shalizi and J. P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *J. Stat. Phys.*, 104:817–879, 2001.

[5] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, New York, 1995.

[6] N. Travers and J. P. Crutchfield. Asymptotic synchronization for finite-state sources. 2010. in preparation.

[7] C. J. Ellison, J. R. Mahoney, and J. P. Crutchfield. Prediction, retrodiction, and the amount of information stored in the present. *J. Stat. Phys.*, 136(6):1005–1034, 2009.

[8] J. P. Crutchfield, C. J. Ellison, and J. R. Mahoney. Time's barbed arrow: Irreversibility, crypticity, and stored information. *Phys. Rev. Lett.*, 103(9):094101, 2009.

[9] J. R. Mahoney, C. J. Ellison, and J. P. Crutchfield. Information accessibility and cryptic processes. *J. Phys. A: Math. Theo.*, 42:362002, 2009.

[10] D. P. Feldman and J. P. Crutchfield. Synchronizing to periodicity: The transient information and synchronization time of periodic sequences. *Advances in Complex Systems*, 7(3-4):329–355, 2004.

[11] J. P. Crutchfield and D. P. Feldman. Regularities unseen, randomness observed: Levels of entropy convergence. *CHAOS*, 13(1):25–54, 2003.

[12] N. Travers and J. P. Crutchfield. Exact synchronization for finite-state sources. 2010. arXiv:1008.4182 [nlin.CD].

[13] T. H. Corment, c. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, third edition, 2009.

[14] B. D. Johnson, J. P. Crutchfield, C. J. Ellison, and Carl S. McTague. Enumerating finitary processes. 2010. in preparation.

[15] Unifilar is known as "deterministic" in finite automata literature. Here, we avoid that term so that confusion does not arise due to the stochastic nature of the models being used.

[16] An ϵ-machine consists of transient and recurrent states, but our focus here is only on the recurrent states, unless otherwise stated.